Encryption Beyond Group Homomorphism Bilinear Groups

Lecture 18

Recall

Homomorphic Encryption

- Group Homomorphism: Two groups G and G′ are homomorphic if there exists a function (homomorphism) $f:G \to G'$ such that for all $x,y \in G$, $f(x) +_{G'} f(y) = f(x +_G y)$
- Homomorphic Encryption: A CPA secure (public-key) encryption s.t. $\mathrm{Dec}(C) +_M \mathrm{Dec}(D) = \mathrm{Dec}(C +_C D)$ for ciphertexts C, D
  - i.e. $\mathrm{Enc}(x) +_C \mathrm{Enc}(y)$ is like $\mathrm{Enc}(x +_M y)$
  - Interesting when $+_C$ doesn't require the decryption key
- e.g., El Gamal: $(g^{x1},m_1Y^{x1}) \times (g^{x2},m_2Y^{x2}) = (g^{x3},m_1m_2Y^{x3})$
- e.g., Paillier: $g^{m1}r_1^n \times g^{m2}r_2^n = g^{m1+m2}r_3^n$

Homomorphic Encryption

- Ring Homomorphism: Two rings A and A′ are homomorphic if there exists a function (homomorphism) $f:A \to A'$ s.t. $\forall x,y \in A$, $f(x) +_{A'} f(y) = f(x +_A y)$ and $f(x) \times_{A'} f(y) = f(x \times_A y)$
- Fully Homomorphic Encryption: A CPA secure (public-key) encryption s.t. $Enc(x) +_C Enc(y)$ is like $Enc(x +_M y)$ and $Enc(x) \times_C Enc(y)$ is like $Enc(x \times_M y)$
  - Candidate solutions since 2009 using "lattice" problems
  - Today: a simpler kind of encryption, which supports only one multiplication (and any number of additions before and after the multiplication)
    - Uses "bilinear pairings"

Bilinear Pairing

- Two (or three) groups with an efficient pairing operation, $e: G \times G \rightarrow G_T$ that is "bilinear"
  - Typically, prime order (cyclic) groups
  - $e(g^a,g^b) = e(g,g)^{ab}$
    - Multiplication (once) in the exponent!
    - $e(g^a,g^b)\ e(g^{a'},g^b) = e(g^{a+a'},g^b)$; $e(g^a,g^{bc}) = e(g^{ac},g^b)$; ...
  - Not degenerate: $e(g,g,) \neq 1$
- D-BDH Assumption: For random (a,b,c,z), the distributions of (g^a,g^b,g^c,g^{abc}) and (g^a,g^b,g^c,g^z) are indistinguishable

3-Party Key Exchange

- A single round 3-party key-exchange protocol secure against passive eavesdroppers (under D-BDH assumption)
  - Generalizes Diffie-Hellman key-exchange
- Let $e: G \times G \rightarrow G_T$ be bilinear and g a generator of G
- Alice broadcasts g^a, Bob broadcasts g^b, and Carol broadcasts g^c
- Each party computes $e(g,g)^{abc}$
  - e.g. Alice computes $e(g,g)^{abc} = e(g^b,g^c)^a$
  - By D-BDH the key $e(g,g)^{abc} = e(g,g^{abc})$ is pseudorandom given eavesdropper's view (g^a,g^b,g^c)

Some More Assumptions

- Computational-BDH Assumption: For random (a,b,c), given (g^a,g^b,g^c) infeasible to find g^{abc}
- Decision-Linear Assumption: $(h_1,h_2,g,h_1^x,h_2^y,g^{x+y})$ and $(h_1,h_2,g,h_1^x,h_2^y,g^z)$ are indistinguishable
- Strong DH Assumption: For random x, given (g,g^x) infeasible to find $g^{1/x}$ or even $(y,g^{1/(x+y)})$. (Note: can check $e(g^xg^y, g^{1/(x+y)}) = e(g,g)$.)
  - q-SDH: Given $(g,g^x,\ldots,g^{x^q})$, infeasible to find $(y,g^{1/(x+y)})$
- Subgroup-Decision Assumption: Indistinguishability of random elements in G from those in a large subgroup of G (requires G to have composite order)
- DDH when $e:G_1 \times G_2 \rightarrow G_T$: DDH could hold in G_1 and/or G_2

BGN Encryption

- Boneh-Goh-Nissim Encryption scheme
  - Supports one multiplication and any number of additions through a layer of encryption
  - Based on the Subgroup-Decision Assumption
  - e: $G \times G \to G_T$ where G is a cyclic group with a large non-trivial subgroup
    - $|G| = pq$, a product of two (similar-sized) primes
    - $H \subseteq G$ generated by $h=g^q$, where g generates G, has $|H|=p$
    - Assumption: A random element in H are indistinguishable from a random element in G (cf. DCR)

BGN Encryption

- e: $G \times G \rightarrow G_T$ where G is a cyclic group with $|G|=pq$, and Subgroup-Decision assumption holds for $H \subseteq G$, $|H|=p$
- Message space = Ring of integers modulo n
  - But efficient decryption will be provided only for a small subset of messages
  - In fact, correct decryption will be possible only up to G/H (e.g., {0,..,q-1}) even inefficiently
- Idea: $Enc_{g,h}(m;r) = g^m h^r$, where g generates G and $h=g^q$ generates H, so that encrypted messages can be added by multiplying ciphertexts, multiplied by plaintext by exponentiating, and multiplied together by pairing ciphertexts
  - $e(g^{m+qr},g^{m'+qr'}) = g_T^{mm' + qr''}$ where $g_T = e(g,g)$ generates G_T

BGN Encryption

- Key generation: Sample n = pq, G s.t. |G|=n, and generator g for H. Public key includes (G,g,h) and secret-key is (G,g,p).
- $\mathrm{Enc}_{g,h}(m;r) = g^m h^r$, where g generates G and $h=g^q$ generates H
- $\mathrm{Dec}_{g,p}(c)$: Find m s.t. $g^{mp} = c^p$ (by brute force, when m is from a small set)
  - $c^p = g^{mp}h^{rp} = g^{mp}$ since $h^p = g^n = 1$

Quadratic speedup using "Pollard's Kangaroo method" for discrete log

- Homomorphic operations (in group G):
  $c_1 +_C c_2 = c_1 \cdot c_2$, $a*c = c^a$ and $c_1 \times_C c_2 = e(c_1,c_2)$
  - But $\times_C$ results in a ciphertext in G_T! Decryption and homomorphic addition and multiplication by plaintext (but not multiplication of two encrypted values) are defined for these ciphertexts too
- CPA secure under Subgroup-Decision assumption on G and H (which implies the same for G_T and H_T): Encryption using a random element in G instead of h^r (random element in H) has no information about message.

2-DNF Computation using BGN Encryption

- Consider a passive-secure 2-party computation problem where Bob has an input bit-vector x and Alice has a secret "2-DNF formula" f. Bob should get f(x) only, and Alice should learn nothing.
  - Disjunctive Normal Form: OR (disjunction) of ANDs
  - 2-DNF: $\vee_{i=1 \text{ to } n} (y_i \wedge z_i)$ where y_i, z_i are literals (input variables or their negations)
  - Passive-secure protocol:
    - Bob generates keys for BGH encryption, encrypts each bit using it, and sends the PK and ciphertexts to Alice
    - Alice homomorphically computes $c{:=}\mathrm{Enc}(r \cdot f'(x))$ where f' is a degree-2 polynomial version of f, using + for $\vee$ and $\times$ for $\wedge$ and (1-x) for $\neg x$, and r random. Bob can (only) check if $f'(x)=0$ or not.

Full-fledged decryption not needed in the protocol

2-DNF Computation using BGN Encryption

- In some applications, want to protect against encryption of illegal values
- Can protect against revealing information by blinding encrypted outputs
  - Instead of returning a ciphertext c, return c $+_c$ Enc(α), where α=0 if all given values are valid, and random otherwise
    - $\alpha = \sum_{i=1 \text{ to } n} r_i \cdot x_i \cdot (1-x_i)$
    - Enc(α) can be computed from { Enc(x_i) }$_I$

Beyond One Multiplication?

- Instead of bilinear maps, if n-linear maps are available, can support up to degree n polynomials
  - Open problem to construct good candidates for multi-linear maps
- Somewhat Homomorphic Encryption
  - Homomorphic encryption supporting an a priori upper bound on the degree of the polynomials to be evaluated
  - Ciphertexts live at different levels, and multiplication leads to higher levels (say, levels add up)
- Fully Homomorphic Encryption: No a priori bound on the degree of the polynomials that can be homomorphically evaluated